

WHAT IS CLAIMED IS:

- 1        1. A software package verification tool for verifying a software package that  
2           includes at least one software component, the tool comprising:  
3                a framework operable to identify at least one test module defining a test of  
4                at least one parameter of the at least one software component of the  
5                package; and  
6                a control module operable to access the framework to cause the at least  
7                one test module identified therein to perform the test defined thereby for  
8                verifying the package.
- 1        2. The tool of claim 1, wherein the framework identifies a plurality of test  
2           modules.
- 1        3. The tool of claim 2, wherein the framework identifies a priority for each of the  
2           test modules.
- 1        4. The tool of claim 3, wherein the control module is operable to cause the test  
2           modules to be executed sequentially according to the priority identified in the  
3           framework for each of the test modules.
- 1        5. The tool of claim 1, wherein a mechanism is provided for identifying the at  
2           least one test module as being one of active and not active.
- 1        6. The tool of claim 5, wherein the mechanism for identifying the at least one test  
2           modules as being one of active and not active is included in the framework.
- 1        7. The tool of claim 5, wherein the mechanism for identifying the at least one test  
2           modules as being one of active and not active is included in the control  
3           module.
- 1        8. The tool of claim 2, wherein the framework comprises a directory having a

2 plurality of entries, each entry identifying one of the plurality of test modules.

1 9. The tool of claim 8, wherein each entry defines a priority for the one of the test  
2 modules identified therein.

1 10. The tool of claim 8, wherein the identity of the one of the test modules  
2 defines its priority.

1 11. The tool of claim 2, wherein each of the plurality of test modules is formed  
2 by a script and the framework identifies each of the test modules by a name  
3 for the script.

1 12. The tool of claim 2, wherein each of the test modules is formed by a software  
2 object.

1 13. A computer program on a carrier medium for verifying a software package  
2 that includes at least one software component, the computer program  
3 comprising computer executable instructions:  
4 a) forming a framework operable to identify at least one test module  
5 defining a test of at least one parameter of the at least one software  
6 component of the package; and  
7 b) forming a control module operable to access the framework to cause the  
8 at least one test module identified therein to perform the test defined  
9 thereby for verifying the package.

1 14. A program storage device readable by a computer, tangibly embodying a  
2 program of instructions executable by the computer to perform method steps  
3 for verifying a software package that includes at least one software  
4 component, the method comprising the steps of:  
5 a) providing a framework for identifying at least one test module, each  
6 said test module defining a test of at least one parameter of the at least  
7 one software component of the package;

8                   b) accessing the framework to identify the at least one test module; and  
9                   c) causing the at least one test module to perform the test defined thereby  
10                  on the package.

1       15. The method of claim 14, wherein the framework identifies a plurality of the  
2       test modules.

1       16. The method of claim 15, wherein a priority for each of the test modules is  
2       identified in the framework.

1       17. The method of claim 15, comprising sequentially causing each of the test  
2       modules to be executed according to the priority identified for each of the test  
3       modules.

1       18. The method of claim 15, comprising identifying each of the test modules as  
2       being one of active and not active.

1       19. The method of claim 15, comprising providing a directory in the framework,  
2       wherein the directory has a plurality of entries, each entry identifying one of  
3       the plurality of test modules.

1       20. A system for verifying a software package that includes at least one software  
2       components, the system comprising:

3               a) a framework to identify at least one test module defining a test of at  
4               least one parameter of the at least one software components of the  
5               package; and  
6               b) a control module operable to access the framework for causing the at  
7               least one test module identified therein to perform the test defined  
8               thereby for verifying the package.

1       21. The system of claim 20, wherein the system comprises a computer including  
2       a processor, memory and software held in the memory and operable to

3 control the processor, the software forming:  
4 said framework and said control module.

1 22. A computer system for verifying a software package that includes at least  
2 one software component, the system comprising:  
3 a) a memory for storing software; and  
4 b) a processing unit for executing the software to carry out the steps of:  
5 (i) providing a framework to identify at least one test module defining a  
6 test of at least one parameter of the at least one software component of  
7 the package; and  
8 (ii) providing a control module operable to access the framework for  
9 causing the at least one test module identified therein to perform the  
10 test defined thereby for verifying the package.

1 23. A method of verifying a software package that includes at least one software  
2 component, the method comprising:  
3 a) providing a framework for identifying at least one test module, each said  
4 test module defining a test of at least one parameter of the at least one  
5 software component of the package;  
6 b) accessing the framework to identify the at least one test module; and  
7 c) causing the at least one test module to perform the test defined thereby  
8 on the package.

1 24. The method of claim 23, wherein the framework identifies a plurality of  
2 test modules.

1 25 The method of claim 24, wherein a priority for each of the test modules is  
2 identified in the framework.

1 26. The method of claim 25, comprising sequentially causing the test modules  
2 to be executed according to the priority identified for each of the test  
3 modules.

- 1 27. The method of claim 24, comprising identifying each of the test modules as  
2 being one of active and not active.
- 1 28. The method of claim 24, comprising providing a directory in the framework,  
2 wherein the directory has a plurality of entries, each entry identifying one of  
3 the test modules.
- 1 29. The method of claim 28, wherein each entry defines a priority of the test  
2 module identified thereby.
- 1 30. The method of claim 28, wherein the identity of a module defines its  
2 priority.
- 1 31. A method of verifying a software package that includes at least one  
2 software component, the method comprising:  
3 a) receiving the software package;  
4 b) accessing a framework that references at least one test module, each said  
5 test module defining a test of at least one parameter of the at least one  
6 software component of the package, for identifying the at least one test  
7 module from the framework; and  
8 c) performing the test defined by the at least one test module on the  
9 package.
- 1 32. The method of claim 31, including repeating steps (b) and (c) to perform a  
2 sequence of tests, the order in which the tests are performed being  
3 determined by relative priorities assigned to each of the at least test  
4 module.
- 1 33. A computer readable medium having stored thereon a data structure  
2 operable for use in verifying a software package that includes at least one  
3 software component, the data structure comprising:

- 4 a) a first data field containing data representing one of a plurality of test  
5 modules, each test module being operable to test at least one parameter  
6 of the at least one software component of the package,  
7 b) where data representing ones of the test modules may be added to and  
8 deleted from the data structure, creating a flexible test structure.

1 34. The medium according to claim 33, wherein the data structure further  
2 comprises a second data field identifying a priority for each of the test  
3 modules represented by the data in the first data field, the priority defining  
4 an order of execution of the test modules.

1 35. The medium according to the claim 33, wherein the data structure further  
2 comprises a third data field identifying the one of a plurality of test modules  
3 represented by the data in the first data field as being one of active and not  
4 active.